DS ASSIGNMENT NO: 06

Coffee Shop Line (Simple Queue)

Name:- Aagam Gadiya PRN :- B24CE1118

Date:-

PROBLEM STATEMENT: Coffee Shop Line (Simple Queue):

Arrival: Customers arrive at the coffee shop and stand in line. Order Processing: The first customer in line gets their order taken, and the barista starts making the coffee. Serving: Once the first customer is served, they leave the queue, and the next customer in line moves forward to be served. Write a program to implement a simple queue

CODE

```
#include<iostream>
#define SIZE 5
using namespace std;
class coffee_shop {
  public:
     int token_queue[SIZE];
     int rear, front;
     coffee_shop() {
       rear = -1;
       front = -1;
     }
     int full() {
        return (rear == SIZE - 1);
     int empty() {
        return (front == -1 || front > rear);
     }
     bool enqueue(int token_no) {
        if (full()) {
          cout << "Queue is full! Cannot accept new tokens." << endl;
          return false; // clearly indicate failure
```

```
} else {
                  if (rear == -1) // first insertion
                    front = 0;
                  rear++;
                  token_queue[rear] = token_no;
                  return true; // insertion successful
               }
             }
             int dequeue() {
               if (empty()) {
                  cout << "Queue is empty" << endl;
                  return 0;
               } else {
                  int token_no = token_queue[front];
                  front++;
                  return token_no;
               }
             }
             void display() {
               if (empty()) {
                  cout << "Queue is empty!" << endl;
                  return;
               }
               cout << "Current queue: ";
               for (int i = front; i <= rear; i++) {
                  cout << token_queue[i] << " ";
               }
               cout << endl;
             }
       };
        int main() {
          int token_no;
          int choice;
          char c;
          coffee_shop c1;
          do {
             cout << "\n----WELCOME to COFFEE SHOP----\n1. Get token\n2. Process token\n3. Display
queue\n4. Exit" << endl;
             cout << "Enter your choice: ";
```

```
cin >> choice;
  switch(choice) {
     case 1: {
        cout << "Enter token no: ";
        cin >> token_no;
        if (c1.enqueue(token_no)) {
          cout << "Token " << token_no << " added" << endl;</pre>
       } else {
          cout << "Invalid! Queue is full, cannot add token " << token_no << endl;</pre>
        }
        break;
     }
     case 2: {
        token_no = c1.dequeue();
        if (token_no != 0) {
          cout << "Token " << token_no << " processed" << endl;</pre>
       }
        break;
     }
     case 3: {
        c1.display();
        break;
     }
     case 4: {
        cout << "Exiting.....";
        return 0;
     }
     default:
        cout << "Invalid choice! Please select a valid option." << endl;</pre>
        break;
  cout << "Do you want to continue (Y/N)? ";
  cin >> c;
} while (c == 'Y' || c == 'y');
return 0;
```

}

OUTPUT

----WELCOME to COFFEE SHOP--------WELCOME to COFFEE SHOP---- Get token 1. Get token -----WELCOME to COFFEE SHOP---- 2. Process token Display queue Process token 1. Get token Display queue Exit Process token Enter your choice: 1 Display queue Exit Enter your choice: 2 Enter token no: 5 Exit Enter your choice: 1 Token 2 processed Token 5 added Do you want to continue (Y/N)? y Do you want to continue (Y/N)? y Token 1 added ----WELCOME to COFFEE SHOP----Do you want to continue (Y/N)? y -----WELCOME to COFFEE SHOP----1. Get token Get token -----WELCOME to COFFEE SHOP---- 2. Process token Process token Display queue Display queue Get token 4. Exit 4. Exit Process token Enter your choice: 2 Enter your choice: 1 Display queue Token 3 processed Enter token no: 6 4. Exit Enter your choice: 1
Enter token no: 2 Queue is full! Cannot accept new tokens. Do you want to continue (Y/N)? y Invalid! Queue is full, cannot add token 6 ----WELCOME to COFFEE SHOP----Do you want to continue (Y/N)? y Token 2 added Get token Do you want to continue (Y/N)? y Process token ----WELCOME to COFFEE SHOP----Display queue -----WELCOME to COFFEE SHOP---- 1. Get token 1. Get token Exit Process token Enter your choice: 2 Process token Display queue Display queue Token 4 processed 4. Exit Do you want to continue (Y/N)? y Enter your choice: 3 Exit 4. Exit
Enter your choice: 1 Current queue: 1 2 3 4 5 ----WELCOME to COFFEE SHOP----Do you want to continue (Y/N)? y Enter token no: 3 1. Get token Token 3 added Do you want to continue (Y/N)? y -----WELCOME to COFFEE SHOP----Process token Display queue Get token Exit -----WELCOME to COFFEE SHOP---- 2. Process token Enter your choice: 2 Display queue Get token Token 5 processed 4. Exit Process token Display queue Do you want to continue (Y/N)? y Enter your choice: 2 Token 1 processed 4. Exit ----WELCOME to COFFEE SHOP----Do you want to continue (Y/N)? y Enter your choice: 1 Enter token no: 4 Get token ----WELCOME to COFFEE SHOP----2. Process token Token 4 added 3. Display queue Do you want to continue (Y/N)? y 1. Get token 4. Exit Enter your choice: 2 Queue is empty Do you want to continue (Y/N)? y ----WELCOME to COFFEE SHOP----1. Get token 2. Process token 3. Display queue 4. Exit Enter your choice: 3 Queue is empty! Do you want to continue (Y/N)? y ----WELCOME to COFFEE SHOP----1. Get token 2. Process token 3. Display queue

4. Exit

Exiting.....

Enter your choice: 4

...Program finished with exit code 0
Press ENTER to exit console.